REMARKS

In the Office Action mailed March 21, 2005, claims 4, 5, 16, 22, 24, 25, and 41-46 are objected to for various informalities. Claims 1-49 are rejected under 35 USC §102(b) as being anticipated by Kaviani (U.S. Publication No. 2002/0079921 A1).

Response to Objections

In response to the objection to the claims, Applicants have amended the claims to more clearly claim the invention. In particular, Applicants have amended claims 16, 22, 24, 41, and 46 to define the acronyms for look up table (LUT) or look up tables (LUT) in each of the claims. Applicants have also amended claims 25 and 46 to define the acronyms for a programmable logic device (PLD) and a factored cube set (FCS), respectively. These amendments merely expand well-known acronyms, and therefore do not relate to patentability.

In response to the suggestion that the term "p-terms" in claims 4, 5, 25, and 42-45is unclear, Applicants note that a p-term is a product term, as defined in paragraph [0024] of the specification. Referring specifically to the language of paragraph [0024], a product term is a term comprising a product of a plurality of variables. Applicants respectfully submit that the term "p-term" is clearly defined in the specification.

In response to the suggestion that it is unclear what is meant by "softPAL" in claim 25, Applicants respectfully submit that "softPAL" is defined in paragraph [0022] as "a complex function that when written in a sum-of-products (SOP) form is too wide to be implemented in a single LUT but can be implemented using a combination of these LUTs and other special structures." Applicants respectfully submit that the term "softPAL" is clearly defined in the specification.

In response to the suggestion that it is unclear what is meant by the recitation "a combination of LUTs and original sized softPALs" in claim 41, Applicants suggest that the language of the claim is clear in view of the specification. For example, in paragraph [0058], an embodiment having a combination of LUTs and softPALs is described where, once every node is assigned an actual delay, the critical paths in the labeled design are computed. The nodes along each critical path are then re-labeled utilizing larger LUTs or softPALs (if available). This re-labeling of the nodes along each critical path is repeated until delay in the critical path does not improve. In view of the definitions in paragraph [0058], Applicants respectfully submit that the language of the claim is clear.

In response to the suggestion that it is unclear what Applicants intended by "Ana, L[p], R[q], Ano" in claim 43-44, Applicants respectfully submit that support for the language in the claim can be found in paragraphs [0065] and [0067], which indicate that:

> A_{NA} comprises a node array;
> L[p] comprises elements of a node array A_L where $p = 1$ to n;
> R[q] comprises elements of a node array A_R where $q = 1$ to m;
> and
> A_{NO} comprises a current node array.

In view of the definitions in paragraphs [0065] and [0067], Applicants respectfully submit that the terms of the claim are clearly defined.

Finally, in response to the suggestion that it is unclear what is meant by "assigning the node a delay" which is found only in claim 46, Applicants have amended claim 46 to more clearly recite the step as "assigning a delay to the node." Applicants respectfully submit that the claim is more clearly defined. The amendment merely substitutes equivalent language to clarify an awkward phrase, and therefore does not relate to patentability.

Rejections under 35 USC §102

In response to the rejection of claims 1-49 under 35 USC §102(b) as being anticipated by Kaviani, Applicants have amended independent claims 1, 22, 24, 25, 47 and 48 to distinguish over Kaviani. Applicants respectfully submit that the remaining independent claims as pending are allowable over Kaviani, and that each dependent claim is allowable for the same reason that its corresponding independent claim is believed allowable. Applicants will address each independent claim separately to clearly show how each independent claim distinguishes over Kaviani.

Independent Claim 1

In response to the rejection of independent claim 1, Applicants have amended the claim to distinguish over Kaviani. Claim 1 relates to a method for mapping a function to logic in a programmable logic device having at least one look up table (LUT) and at least one dedicated logic element. Applicants have amended the claim to include a step of "placing input variables of the function having later arrival times closer to the output location of the function than input variables having earlier arrival times." Applicants respectfully submit that Kaviani fails to disclose or suggest a method of mapping a function to logic in a programmable logic device wherein input variables having later arrival times are placed closer to the output.

There are references in the Office Action to sections of Kaviani for disclosing factors where signals with the latest arrival times are implemented by logic elements closest to an output location of the function (e.g., page 5 of the Office Action describing a step of implementing the factored form of the function such that each factor is implemented by one of the logic elements). In particular, paragraphs [0017], [0053], [0054], [0068], [0102], and [0203]-[0208] are cited for disclosing signals with the latest arrival times being implemented closest to an output location of the function. However, Applicants respectfully submit that none of these paragraphs discloses or suggests placing input variables of the function having later arrival times closer to the output location of the function than input variables having earlier arrival times as claimed by Applicants. Applicants will address each of these paragraphs separately below.

While paragraph [0017] of Kaviani is unrelated to arrival times of input variables, paragraphs [0053] and [0054] of Kaviani fail to disclose or suggest placing input variables having later arrival times closer to the output location of the function than variables having earlier arrival times as claimed by Applicants. Paragraph [0053] relates to reducing the number of slices required to implement a function, where "reducing the required number of slices improves area efficiency, and the resulting reduction in signal propagation delay improves speed performance." That is, by reducing the number of slices required to implement the function, signal propagation delay is inherently reduced and speed is improved. While paragraph [0054] discloses using dedicated function generators in place of look up table logic to combine slice outputs to improve speed, there is no teaching or suggestion in paragraph [0054] to place input variables having later arrival times closer to the output location of the function than input variables having earlier arrival times.

Similarly, paragraph [0068] relates to a flow chart of Fig. 2 which illustrates the process of decomposing a wide fan-in circuit design expressed in a sum of product (SOP) format. According to paragraph [0068], the various p-terms are sorted in descending order to provide a protocol (referred to in Kaviani as "literal sharing") for combining p-terms which are input to given look up tables. For example, as shown in table 2 of Kaviani, it is determined whether the second p-term P5 (or some other p-term) can be combined with p-term P7. However, there is no teaching or suggestion in paragraph [0068] that input variables having later arrival times are placed closer to the output location of the function than input variables having earlier arrival times. Referring to paragraph [0102], an example of "literal sharing" shows how literal sharing reduces both logic and routing delay, thus increasing the circuit performance. However, there is no teaching or suggestion in paragraph [0102] to place input variables of the function having later arrival times closer to the output location of the function than input variables having earlier arrival times.

Finally, paragraphs [0203]-[0208] refer to the selection and use of horizontal and vertical expanders or SOP expanders. While horizontal expander chains (as described in Fig. 14) can be used to enable a greater number of inputs for a p-term, a vertical expander chain can then be used to combine a large number of p-terms to generate a wide PAL output signal. As shown in FIG. 22, several vertically-positioned CLEs can be combined by using the configuration of FIG. 21, but extending the vertical expander chain across CLE boundaries. The vertical expander chain can be made as long as necessary to implement any size of PAL logic. However, an extremely long expander chain would be very slow. Therefore, for very wide functions, the CLE of FIG. 14 provides a second type of horizontal expander chain--the SOP chain--that can be used to combine the outputs of several vertical expander chains. Accordingly, the disclosure in paragraphs [0203]-[0208] relate to a different configuration of slices and combinatorial logic to implement wide functions. However, there is no teaching or suggestion in paragraphs [0203]-[0208] to place input variables of the function having later arrival times closer to the output location of the function than input variables having earlier arrival times, as claimed by Applicant. Accordingly, Applicants submit that independent claim 1 and its dependent claims are allowable over Kaviani.

Independent Claim 22

In response to the rejection of independent claim 22, Applicants have amended the claim to distinguish over Kaviani. In particular, claim 22 is directed to a programmable logic device configured to implement a function in factored form having a plurality of factored cube sets, wherein each factored cube set comprises a shared set and an unshared set. Claim 22 comprises a first dedicated logic element coupling the first LUT and the second LUT to form a first LUT chain. Applicants have amended claim 22 to indicate that the input variables of the function having later arrival times are placed closer to the output location of the function than input variables having earlier arrival times. Applicants respectfully submit that Kaviani fails to disclose or suggest a programmable logic device configured to implement a function in factored form having a plurality of factored cube sets, wherein input variables of the function having later arrival times are placed closer to the output location of the function than input variables having earlier arrival times, for the same reasons as set forth above with respect to claim 1. Accordingly, Applicants submit that independent claim 22 and its dependent claims are allowable over Kaviani.

Independent Claim 24

In response to the rejection of independent claim 24, Applicants have amended the claim to distinguish over Kaviani. In particular, claim 24 is also directed to a programmable logic device configured to implement a function in factored form having a plurality of factored cube sets. The programmable logic device of claim 24 as amended comprises a first dedicated logic element coupling the first combination of LUTs and the second combination of LUTs to form a first chain of combinatorial look up table (LUT) elements, wherein "input variables of the function having later arrival times are placed closer to the output location of the function than input variables having earlier arrival times." Applicants again submit that Kaviani fails to disclose or suggest a programmable logic device, configured to implement a function in factored form having a plurality of factored cube sets, wherein input variables of the function having later arrival times are placed closer to the output location of the function than input variables having earlier arrival times, for the same reasons as set forth above with respect to claim 1. Accordingly, Applicants also submit that independent claim 24 and its dependent claims are allowable over Kaviani.

Independent Claim 25

In response to the rejection of independent claim 25, Applicants have also amended the claim to distinguish over Kaviani. In particular, claim 25 is directed to a method of mapping a softPAL into a programmable logic device (PLD), and comprises a step of implementing the at least one chain of FCSs with a combination of LUTs and dedicated resources. Applicants have amended the claim to indicate that the step of implementing the at least one chain of FCSs comprises placing input variables of the function having later arrival times closer to the output location of the function than input variables having earlier arrival times. Applicants respectfully submit that Kaviani fails to disclose or suggest a method of mapping a softPAL into a programmable logic device (PLD) comprising a step of implementing the at least one chain of FCSs with a combination of LUTs and dedicated resources, wherein input variables of the function having later arrival times are placed closer to the output location of the function than input variables having earlier arrival times. Accordingly, Applicants submit that independent claim 25 and its dependent claims are allowable over Kaviani.

Independent Claim 36

In response to the rejection of independent claim 36, Applicants respectfully submit that the claim as pending clearly distinguishes over Kaviani. In particular, independent claim 36 relates to a method for mapping a function to logic elements in a programmable logic device, and comprises steps of:

> determining arrival time for input signals to each factor in the factored form; and
>
> implementing the factored form of the function such that each factor is implemented by one of the logic elements,

> and factors having signals with the latest arrival times are implemented by logic elements closest to an output location of the function.

Applicants respectfully submit that Kaviani fails to disclose or suggest either step. It is suggested in the Office Action that the step of determining arrival times for input signal to each factor in the factored form is disclosed in paragraphs [0053], [0068] or [0102]. However, as described above with respect to claim 1, paragraph [0053] relates to reducing the number of slices required to implement a function to improve area efficiency. Similarly, paragraph [0068] relates to a protocol for "literal sharing" by sorting the various p-terms in descending order to enable combining p-terms which are input to given look up tables. Finally, paragraph [0102] refers to an example of literal sharing. However, none of the paragraphs discloses or suggests determining arrival times for input signals to each factor in the factored form as claimed by Applicants.

Further, there is no teaching or suggestion to implement the factored form of the function such that each factor is implemented by one of the logic elements, and factors having signals with the latest arrival times are implemented by logic elements closest to an output location of the function, for the same reasons set forth above with respect to claim 1. Accordingly, Applicants submit that claim 36 as pending clearly distinguishes over Kaviani, and that independent claim 36 and its dependent claims should be allowed.

Independent Claim 41

In response to the rejection of independent claim 41, Applicants respectfully submit that the claim as pending clearly distinguishes over Kaviani. In particular, claim 41 is directed to a post-mapping optimizing method comprising steps of:

> (a) forming a mapped design mapped to a combination of look up tables (LUTs) and original sized softPALs;
> (b)calculating original critical path delay in the mapped design;
> (c) for each critical path in the mapped design:
> re-mapping the critical path node to a larger sized softPAL;
> if delay is reduced, accepting the large sized softPAL;

> if delay is not reduced, retaining the look up table (LUT) or original sized softPAL;
>
> (d) re-calculating a new critical path delay for a new mapped design;
>
> (e) if the new critical path delay for the mapped design is less than the original critical path delay, accepting the new mapped design; and
>
> (f) repeating steps (a) through (e) until the new critical path delay for the mapped design is not less than the original critical path delay.

Applicants respectfully submit that Kaviani fails to disclose or suggest any of the steps (b) – (f) of claim 41. In particular, Kaviani fails to disclose or suggest a step of calculating an original critical delay path in the mapped design (step b), re-mapping the critical path to a larger softPAL for each critical path in the mapped design (step c), and re-calculating the critical path delay (step d). It is suggested that these steps are disclosed in paragraphs [0204]-[0208], [0142] and [0204]-[0208], and [0074], respectively. However, paragraphs [0204]-[0208] relate to the selection of horizontal and vertical expanders, or the use of sum of product (SOP) expanders if the expander chain becomes extremely long. Furthermore, neither paragraphs [0142] nor [0204]-[0208] relate to re-mapping the critical path node to a larger sized softPAL. That is, once the inputs to the slices of Fig. 22 of Kaviani are selected, a critical path is not recalculated to a larger size softPAL. In contrast, any application of the teaching of paragraphs [0142] and [0204]-[0208] would lead to the selection of different logic to combine outputs of look up tables, but not to re-mapping the critical path node to a larger sized softPAL. Finally, paragraph [0074] relates to determining whether a product term can be input to a given group of look up tables which is coupled to receive one or more other p-terms. The determination of whether a given p-term can be input to a group of look up tables does not disclose creating an original critical path or re-calculating a new critical path delay for a new mapped design. Accordingly, Applicants submit that claim 41 as pending distinguishes over Kaviani, and that claim 41 and its dependent claims should be allowed.

Independent Claim 42

In response to the rejection of independent claim 42, Applicants respectfully submit that the claim as pending clearly distinguishes over Kaviani. In particular, claim 42 relates to a p-term estimation method for a function comprising:

> decomposing the function into a function tree of AND nodes, OR nodes, and NOT nodes where each of the nodes has no more than two inputs;
>
> associate all inputs to a node with an array, where the size of the array is a number of p-terms corresponding to an SOP function performed at an input to the node, and a value of each element of the array is a number of variables input to each p-term of the SOP function; and
>
> computing the array associated with an output of the node based on a function performed by the node.

Applicants respectfully submit that Kaviani does not disclose any p-term estimation method, and fails to disclose or suggest any of the steps of claim 42. Applicants claim decomposing the function into a function tree of AND nodes, OR nodes, and NOT nodes where each of the nodes has no more than two inputs. There is no teaching in Kaviani of decomposing the function into a function tree where each of the nodes has no more than two inputs. Paragraphs [0055], [0081], and [0190] are cited for disclosing decomposing the function into a function tree. However, if the equation (EQ.1) in paragraph [0055], which discloses the generation of a sum of products, were mapped to a function tree, the resulting tree would clearly have both AND nodes and OR nodes having more that two inputs. Similarly, paragraph [0081] relates to protocol for determining whether a product term can fit into a given product chain based upon a circuit having a 4 input lookup table. The step of paragraph [0081] relates to a circumstance when the carry chain of a slice is used as an OR gate, and the composite number of literal inputs to the row and the look up table is less than or equal to 4. Accordingly, paragraph [0081] clearly does not disclose decomposing the function into a function tree where each of the nodes has no more than two inputs.

Kaviani also fails to disclose a step of associating all inputs to a node with an array, where the size of the array is a number of p-terms of an SOP function performed at an input to the node, and a value of each element of the array is a number of variables input to each p-term of the SOP function. It is suggested in the

Office Action that the step of associating inputs to a node with an array is disclosed in paragraphs [0064], [0078], [0085]-[0086]. However, paragraph [0064] relates to a table having rows associated with various p-terms and the values for input lines for each of the rows. Paragraph [0078] discloses implementing chain P7 and p-term P5 in a single slice 320, as described in reference to Fig. 3B. Finally, paragraphs [0085] and [0086] relate to determining whether a p-term can be combined with a given chain, as described in reference to Fig. 3A. However, none of the paragraphs discloses or suggests associating all inputs to a node with an array as defined in claim 42. Rather, Table 2 of Kaviani relates to various p-terms and required inputs associated with the p-terms.

Finally, Kaviani fails to disclose or suggest computing the array associated with an output of the node based on a function performed by the node. Paragraphs [0054] and [0064] disclose computing a sum of products, and not computing an array associated with the output of a node based upon the function performed by the node as claimed. Applicants respectfully submit that none of the elements of claim 42 are disclosed or suggested by Kaviani, and that claim 42 and its dependent claims should be allowed.

Independent Claim 46

In response to the rejection of independent claim 46, Applicants respectfully submit that the claim as pending clearly distinguishes over Kaviani. In particular, claim 46 relates to a method for determining a delay of a node for implementing a softPAL function, and includes steps of:

> forming a first set of one or more factored cube set (FCS) chains representing the equation;
>
> implementing the first set of FCS chains with a first implementation scheme;
>
> assigning a delay to a node of the first implementation scheme;
>
> forming a second set of one or more FCS chains representing the equation;
>
> implementing the second set of FCS chains with a second implementation scheme; and

> updating the delay of the node when a delay of the second implementation scheme is less than the delay of the node.

Applicants respectfully submit that Kaviani fails to disclose or suggest any of the steps set forth above. Among other differences, Kaviani fails to disclose or suggest determining a delay of a node for implementing a softPAL function, and more importantly, implementing first and second sets of FCS chains according to first and second implementation schemes. It is suggested in the Office Action that the step of forming a first set of one or more factored cube set (FCS) chains representing the equation is disclosed in paragraphs [0085]-[0086], and that the step of forming a second set of one or more FCS chains representing the equation is disclosed in paragraphs [0097]-[0098]. While paragraphs [0085]-[0086] relate to determining a sum of product equation, paragraphs [0097]-[0098] relate to implementing portions of the equation in second and third slices (420 and 430) of Fig. 4A. The cited paragraphs fail to disclose or suggest implementing first and second sets of FCS chains according to first and second implementation schemes. That is, Fig. 4A is a sample combination logic circuit, and does not relate to a second implementation scheme. Kaviani also fails to disclose or suggest updating the delay of the node when a delay of the second implementation scheme is less than the delay of the node. While Kaviani discloses using different combinatorial logic (in the form of horizontal and vertical expanders and SOP expanders) to enable a given set of FCS chains, Kaviani does not disclose or suggest updating the delay of the node when a delay of the second implementation scheme is less than the delay of the node. Accordingly, Applicants submit that independent claim 46 should be allowed.

Independent Claim 47

In response to the rejection of independent claim 47, Applicants have amended claim 47 to distinguish over Kaviani. Claim 47 is directed to a method for decreasing delay of a node in a critical path during a mapping of a function. Applicants have amended claim 47 to include a step of "selecting a second element of a cost table to map the function, the second element of the cost table comprising a different number of inputs than the first element of the cost table." Applicants respectfully submit that

Kaviani fails to disclose or suggest such a method of decreasing the delay of a node in a critical path. Paragraphs [0204]-[0208] are again cited for disclosing each of the elements of the claim. However, paragraphs [0204]-[0208] relate to the selection of horizontal and vertical expanders, or the use of SOP expanders if the expander chain becomes extremely long. Paragraphs [0204]-[0208] fail to disclose or suggest selecting a second element of a cost table to map the function where the second element of the cost table comprises a different number of inputs than the first element of the cost table, as set forth in the amended claim. Accordingly, Applicants submit that claim 47 as amended clearly distinguishes over Kaviani and should be allowed.

Independent Claim 48

In response to the rejection of independent claim 48, Applicants have amended the claim to distinguish over Kaviani. In particular, claim 48 relates to a method of mapping a function to a combination of one or more LUTs and softPALs. Applicants have amended the step of coupling the one or more LUTs with the one or more vertical elements to indicate that input variables having later arrival times are placed closer to the output location of the function than variables having earlier arrival times. Applicants respectfully submit that Kaviani fails to disclose or suggest placing input variables of the function having later arrival times closer to the output location of the function than input variables having earlier arrival times, for the same reasons as set forth above with respect to claim 1. Accordingly, Applicants submit that independent claim 48 should be allowed.

Independent Claim 49

In response to the rejection of independent claim 49, Applicants respectfully submit that the claim as pending clearly distinguishes over Kaviani. In particular, claim 49 relates to a method for estimating a number of product terms in a function tree for a node, and comprises steps of:

> receiving a two-bounded function tree;
> determining an array for each child of the node; and

> estimating the number of product terms for the node from the arrays of each child of the node based on a type of the node.

Applicants respectfully submit that Kaviani fails to disclose or suggest estimating a number of product terms in a function tree for a node, and in particular a two-bounded function tree as claimed by Applicants. It is suggested in the Office Action that the step of determining an array for each child of a node is disclosed in paragraphs [0197]-[0200]. However, these paragraphs relate to implementing large PALs, and in particular to the use of different expanders, such as horizontal, vertical or SOP expanders. There is no teaching or suggestion of determining an array for each child of the node, or estimating the number of product terms for the node from the arrays of each child of the node based on a type of the node. Applicants submit that the claim as pending is clearly allowable over Kaviani.

For the reasons set forth above, Applicants submit that the claims as amended are allowable over the cited art and respectfully requests reconsideration.

Respectfully submitted,

Justin Liu
Attorney for Applicants
Reg. No.: 51,959

I hereby certify that this correspondence is being deposited with the United States Postal Service as first-class mail in an envelope addressed to: Commissioner for Patents, P.O. BOX 1450, Alexandria, VA 22313-1450, on June 21, 2005.

Julie Matthews
Name

Julie Matthews
Signature